МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Балтийский федеральный университет имени Иммануила Канта»

ОНК «Институт высоких технологий»

ОТЧЁТ О ПРОХОЖДЕНИИ

УЧЕБНО-ТЕХНОЛОГИЧЕСКОЙ (ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ) ПРАКТИКИ

на базе Высшей школы компьютерных наук и прикладной математики

Выполнила ФИО \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

студентка очной формы обучения 1 курса

направления подготовки 01.03.02 Прикладная математика и информатика

профиль обучения «Информатика и программирование»

Руководитель практики

Ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тарачков М.В..

г. Калининград 2023 г.

Оглавление

[Введение 3](#_Toc137483912)

[Глава 1. Базовые понятия Языка C++ 4](#_Toc137483913)

[Базовые типы данных. 4](#_Toc137483914)

[Условные операторы и циклы 5](#_Toc137483915)

[Массивы и указатели. Функции. 6](#_Toc137483916)

[Глава 2. Задания на практику 7](#_Toc137483917)

[Задача №1 «Шифрование» 7](#_Toc137483918)

[Задача №2 «Быстрое копирование» 7](#_Toc137483919)

[Задача №3 «Обратная кинематика» 8](#_Toc137483920)

[Задача №4 «Умный дом» 9](#_Toc137483921)

[Глава 3. Выполнение заданий на практику 11](#_Toc137483922)

[Решение задачи №1 11](#_Toc137483923)

[Решение задачи №2 11](#_Toc137483924)

[Решение задачи №3 11](#_Toc137483925)

[Решение задачи №4 11](#_Toc137483926)

[Заключение 12](#_Toc137483927)

[Список литературы 13](#_Toc137483928)

[Приложения 14](#_Toc137483929)

[Приложение 1 14](#_Toc137483930)

[Приложение 2 14](#_Toc137483931)

**Введение**

Вид практики – Учебная технологическая (проектно-технологическая) практика (далее Учебная практика).

Цель учебной практики: получение первичных профессиональных умений навыков.

Задачи учебной практики:

* Закрепление и углубление теоретических знаний по программированию;
* Изучение основ объектно-ориентированного программирования в C++;
* Работа с памятью;
* Работа с библиотеками;
* Приобретение и развитие первичных профессиональных навыков и умений по прикладной математике и информатике.

**Глава 1. Базовые понятия Языка C++**

**Базовые типы данных.**

**Тип переменной** – характеристика, определяющая формат представления данных в памяти компьютера, множество допустимых значений этих данных и совокупность операций над ними. Зная тип переменной, компилятор выделит для нее необходимое количество ячеек памяти и будет знать, какого рода данные будут храниться в этой переменной.

|  |  |  |
| --- | --- | --- |
| Тип | Размер  (в байтах) | Диапазон |
| unsigned char | 1 | 0 – 255 |
| char | 1 | -128 – 127 |
| unsigned short | 2 | 0 – 65535 |
| short | 2 | -32 768 – 32 767 |
| unsigned int (16 разрядов) | 2 | 0 – 65 535 |
| unsigned int (32 разряда) | 4 | 0 – 4 294 967 295 |
| int (16 разрядов) | 2 | -32 768 – 32 767 |
| int (32 разряда) | 4 | -2 147 483 648 – 2 147 483 647 |
| unsigned long | 4 | 0 – 4294967295 |
| long | 4 | -2 147 483 648 – 2 147 483 647 |
| unsigned long long (С99) | 8 | 0 – 18 446 744 073 709 551 615 |
| long long (С99) | 8 | -9 223 372 036 854 775 808 –  9 223 372 036 854 775 807 |
| float | 4 | 1.2e-38 – 3.4e+38 |
| double | 8 | 2.2e-308 – 1.7e+308 |
| long double | 10 | 1.7e-4932 – 1.7e+4932 |

## Условные операторы и циклы

**Условный оператор** позволяет проверить некоторое условие и в зависимости от результатов проверки выполнить то или иное действие. Структура условного оператора имеет следующий вид:

If(выражение) оператор; else оператор;

**Оператор for** формально можно записать следующим образом: for(выражение\_1; выражение\_2; выражение\_3) тело цикла

**Оператор while** имеет форму

While(условие) оператор-или блок;

**Оператор do-while** имеет форму:

Do оператор-или блок; while(условие);

Выражение в скобках может принимать либо нулевое (ложное), либо ненулевое (истинное) значение. Тело цикла будет выполняться, только если выражение истинно. Если же выражение ложно, то цикл завершается. Например, вывод на экран чисел 1, 2, …, 10 можно осуществить следующим образом:

Тело цикла составляет либо одна инструкция, либо набор инструкций, заключенных в фигурные скобки. Выражение\_1 обычно используется для инициализации начальных значений переменных, которые управляют циклом (управляющие переменные). Выражение\_2 определяет условия, при которых тело цикла будет выполняться. Выражение\_3 определяет изменение значений управляющих переменных. Цикл for выполняется по следующей схеме:

1. Вычисляется выражение\_1.

2. Вычисляется выражение\_2.

3. Если значение выражения\_2 является истинным (ненулевым), тогда выполняется тело цикла, а затем вычисляется выражение\_3 и осуществляется переход к выражению\_2. Если же значение выражения\_2 является ложным (нулевым), тогда выполнение оператора for завершается. При отсутствии выражения\_2 оно подразумевается всегда истинным.

Оператор break позволяет выйти из операторов цикла for, while, do-while и переключателя switch. Данный оператор дает возможность немедленного выхода из самого внутреннего цикла или переключателя.

Оператор continue вызывает преждевременное завершение выполнения тела цикла с переходом к следующему шагу итерации. Для циклов while и do-while это означает немедленный переход к проверке условия, а для цикла for – переход к выражению\_3. Оператор continue действует на самый внутренний цикл.

## Массивы и указатели. Функции.

В отличие от обычных переменных, которые хранят одно значение, массивы используются для хранения целого набора однотипных значений. Массив – набор однотипных элементов, последовательно располагающихся в памяти компьютера. Для определения массива необходимо указать тип элементов, которые в него входят, и максимальное количество элементов, которое может быть помещено в массив.

Указатели представляют собой переменные, значениями которых являются адреса памяти. Если в "обычной" переменной содержится некоторое значение, то указатель содержит адрес той или иной переменной. "Обычная" переменная *непосредственно* ссылается на значение, указатель же ссылается на значение *косвенно*.

Синтаксис создания указателя имеет следующий вид: тип \*имя\_переменной,

где тип – тип переменной, адрес которой будет содержаться в указателе, имя\_переменной – имя переменной типа указатель, символ "звездочка" означает, что объявляемая переменная является указателем.

Функция – подпрограмма, которая организовывает тот или иной алгоритмический процесс и возвращает некоторое значение. Про- грамма на языке Си имеет по крайней мере одну функцию main(), которая автоматически вызывается при запуске программы. Сама функция main() может вызывать другие функции, а те, в свою очередь, сами могут вызывать те или иные функции.

Функция состоит из объявлений и операторов и предназначена для решения определенной задачи. Каждая функция должна иметь имя, которое используется для ее объявления, определения и вызова.

Когда имя функции встречается в программе, то управление передается к телу данной функции, то есть осуществляется вызов функции. При вызове функции ей при помощи аргументов (формальных параметров) передаются некоторые значения (фактические параметры), которые используются функцией во время ее работы. Любой аргумент функции может быть константой, переменной или выражением.

Тип возвращаемого значения объявляет тип значения, возвращаемого функцией. Если же функция не возвращает значения, тогда в качестве типа возвращаемого значения выступает пустой тип – void. Если функция не получает аргументы, список параметров также объявляется как void.

Функции могут возвращать значения. После обращения к функции она выполняет некоторые действия и в качестве результата своей работы может возвратить некоторое значение. С помощью инструкции return происходит возврат результата от вызываемой функции к вызывающей. После слова return может следовать любое выражение:

return выражение;

# Глава 2. Задания на практику

**Задача №1 «Шифрование»**

Реализуйте класс, который будет выполнять шифрование и дешифрование сообщения на английском языке. Длина сообщения не более 30 слов. Словом называется последовательность символов до первого пробела. Знак препинания после символов тоже входит в слово.

Ключом является последовательность Фибоначчи.

**Пример**

Есть сообщение

*Hello, my name is Ivan.*

Поставим в соответствие каждому слову в сообщении число из последовательности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hello, | my | name | is | Ivan. |
| 1 | 2 | 3 | 5 | 8 |

Пусть ключ для этого сообщения 2, 8, 5, 1, 3, тогда зашифрованное сообщение примет вид

*my Ivan. is Hello, name*

Для расшифровки сообщения, нужно слова переставить так, чтобы получилась верная последовательность Фибоначчи.

**Реализация**

В классе реализуйте методы Encode и Decode, предусмотрите возможность ввести и вывести ключ, а также обработку ошибок при некорректном вводе.

Помните, что вашим классом могут пользоваться другие разработчики, оставьте публичными только необходимые для работы методы.

Продемонстрируйте работу с классом.

**Задача №2 «Быстрое копирование»**

Вам даны два массива длиной N (N – разумно большое число). Тип данных может быть произвольным. Вам необходимо найти способ скопировать содержимое одного массива в другой наиболее эффективно. Библиотечные функции использовать нельзя.

Такая реализация подойдет, но не во всех случаях:

const int N = 1000;  
some\_type mas\_a[N] = {1,2,3,4,5};  
some\_type mas\_b[N] = {0};  
  
for(int i=0;i<N;++i)  
 mas\_b[i] = mas\_a[i];

Почему? Ну и само собой с вас объяснение, почему ваш код будет копировать эффективно.

**Задача №3 «Обратная кинематика»**

Предположим, у вас есть робот (рис. 1), и вы хотите научить его рисовать.

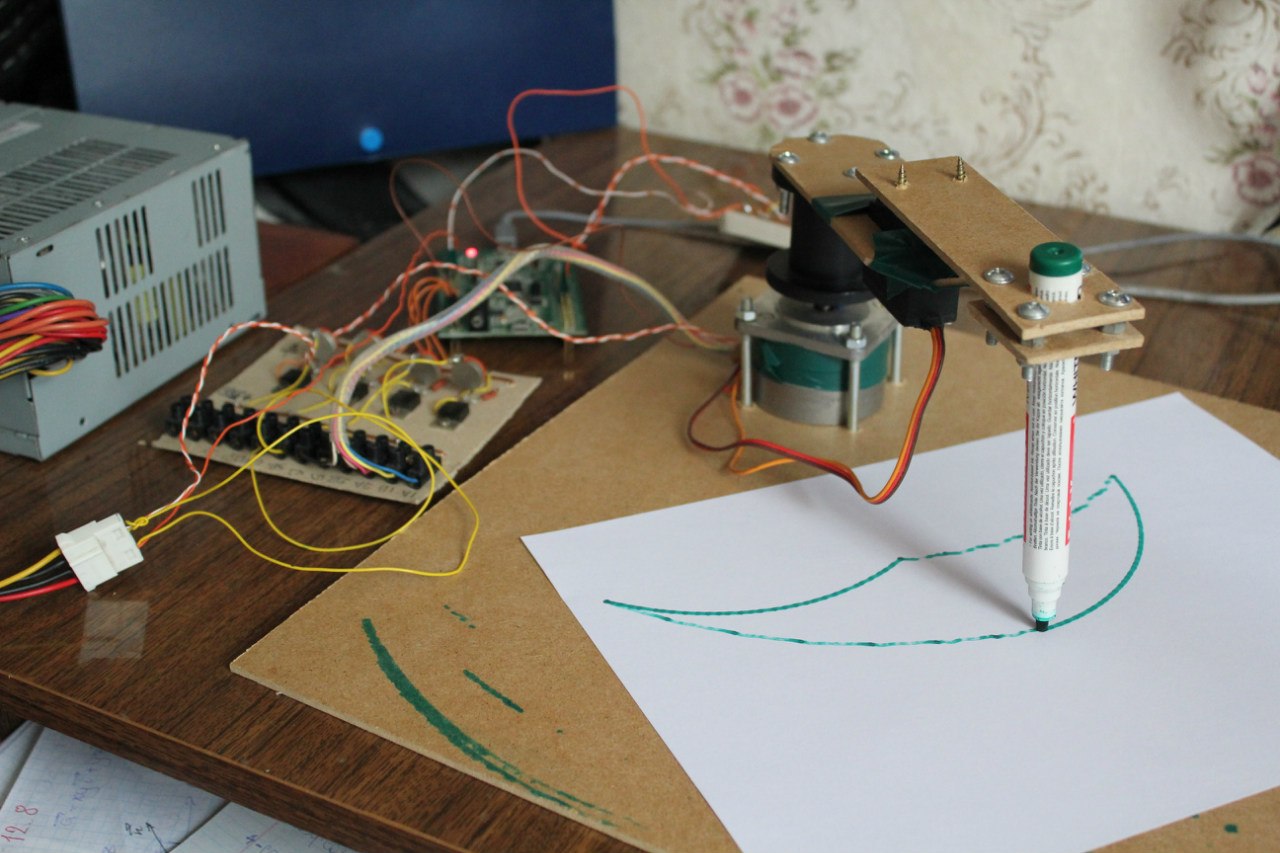


Рис. 1. Робот для рисования.

У робота есть 2 шарнира (joints), в которых установлены электродвигатели. Пусть угол поворота первого двигателя – alpha, второго – beta. Двигатели соединены звеном (link) длиной L1. Второе звено длиной L2 соединяет двигатель и фломастер, он же рабочий орган робота (end effector). Про такого робота говорят, что он имеет 2 степени свободы (degree of freedoms).

Построим математическую модель такого робота. Так как фломастер движется только в плоскости, можно работать в двумерном пространстве.

Введем систему координат OXY. Оба звена представим в виде отрезков. Шарниры и рабочий орган – точки (рис. 2).

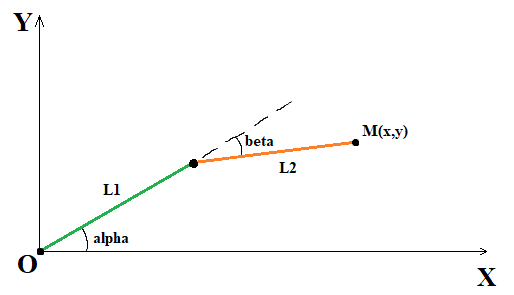


Рис. 2. Схематичное изображение робота.

Особенно важна точка M(x,y) – это координата пишущей части фломастера.

Любая фигура может быть представлена некоторым конечным набором точек Mi, i = 1..n. В каждую из этих точек необходимо последовательно переместить фломастер.

Положение фломастера регулируется углами alpha и beta, чтобы их рассчитать при известной точке M(x,y), применяется задача обратной кинематики (inverse kinematics).

\*тут должно быть продолжение мат. модели, но я не хочу дублировать статьи <https://habr.com/ru/post/358798/> и <https://robocraft.ru/mechanics/756>\*

Реализуйте класс KinematicSolver. Предусмотрите возможность задания длин звеньев L1 и L2, получения углов alpha и beta, функцию Solve(x,y), которая будет рассчитывать углы по заданным координатам.

Продемонстрируйте ваше решение. Чтобы не показывать расчеты на бумаге, можно сделать визуализацию при помощи SFML.

Совет: сначала решите задачу аналитически на бумаге, затем запрограммируйте.

**Задача №4 «Умный дом»**

В системе «Умный дом» используется управляющий компьютер, который взаимодействует с различными приборами.

Приборы подразделяются на несколько категорий:

1. Счетчики электрической энергии
   1. Меркурий 230
   2. Нева МТ314
   3. Энергомера CE308
2. Блоки ввода дискретных сигналов:
   1. Reallab NL-16HV
   2. Приборэлектро PRE-16
   3. Энергосервис ЭНМВ-1-24
3. Блоки управления отоплением
   1. Ouman S203
   2. Овен ТРМ232

Получить данные от каждого прибора можно через метод

*void poll() {*

*std::cout << “Device name” << std::endl;*

*// Некоторая реализация получения данных, которую вам делать  
// не нужно.*

*}*

Для реализации конкретного проекта умного дома проектировщики могут выбрать, как правило, по одному прибору каждого типа. Тем не менее, программа должна поддерживать их все.

Предложите реализацию программы, которая могла бы считывать из файла названия приборов и формировать конфигурацию с учетом конкретного проекта.

Продемонстрируйте, как вы будете вызывать метод poll.

Пишите программу с учетом того, что в будущем количество прибором может возрасти.

**Глава 3. Выполнение заданий на практику**

## Решение задачи №1

Ваше решение, что использовали, какие были сложности

Например,

Был создан класс Chipher, он содержит методы Encode() для кодирования и Decode() для декодирования ….

Код программы расположен в приложении №1

## Решение задачи №2

Ваше решение, что использовали, какие были сложности

Например,

Был создан класс Chipher, он содержит методы Encode() для кодирования и Decode() для декодирования ….

Код программы расположен в приложении №2

## Решение задачи №3

Ваше решение, что использовали, какие были сложности

Например,

Был создан класс Chipher, он содержит методы Encode() для кодирования и Decode() для декодирования ….

Код программы расположен в приложении №3

## Решение задачи №4

Ваше решение, что использовали, какие были сложности

Например,

Был создан класс Chipher, он содержит методы Encode() для кодирования и Decode() для декодирования ….

Код программы расположен в приложении №4

# Заключение

В ходе практики были изучены основы языка программирования С++. Задачи были направлены на закрепление теоретического материала по типам данных, арифметическим, логическим операторам. Были изучены условия и циклы. Особое внимание уделялось работе с текстом, в частности массиву char.

Предлагалось решить задачи, связанные со спортивным программированием, длинной арифметикой, битовыми операциями.

В результате практики были усовершенствованы мои компетенции, закреплены теоретические навыки.

В ходе учебной практики я научилась: выполнять поставленную задачу, следуя условию, изменять готовое решения по мере нахождения ошибок, искать альтернативные пути решения задач.

Также я освоила новую для меня среду программирования Visual Studio и работу с GitHub (создание репозиториев для выгрузки решений задач). Я научилась использовать функции и циклы, обрабатывать строки, изучила библиотеки fstream (для чтения данных из файла и ввода в него полученных результатов), iostrem (для ввода и вывода данных), vector (для работы с динамическим массивом), string (для работы со строками), выполнила работу с разными видами массивов и приемами работы с ними.

По мере прохождения учебно-технологической практики я выполнила пять задач, в которых использовала полученные знания и закрепила навыки работы с C++.

В течение практики задачи были выполнены, а цели достигнуты.

# Список литературы

**Перечень учебной литературы ресурсов сети «Интернет», необходимой для проведения практики**

1. Варфоломеева, Т. Н. Структуры данных и основные алгоритмы их обработки : учебное пособие / Т. Н. Варфоломеева. - Москва : ФЛИНТА, 2017. - 159 с. - ISBN 978-5-9765-3691-3. - Текст : электронный. - URL: https://znanium.com/catalog/product/1860018 (дата обращения: 19.01.2023). – Режим доступа: по подписке.
2. Гданский, Н. И. Основы теории и алгоритмы на графах : учебное пособие / Н. И. Гданский. — Москва : ИНФРА-М, 2020. — 206 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-014386-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/978686 (дата обращения: 19.01.2023). – Режим доступа: по подписке.
3. Затонский, А. В. Программирование и основы алгоритмизации. Теоретические основы и примеры реализации численных методов: учебное пособие / А.В. Затонский, Н.В. Бильфельд. — 2-е изд. — Москва: РИОР : ИНФРА-М, 2022. — 167 с. — (Высшее образование). — DOI: https: //www.dx.doi.org/10.12737/20468. - ISBN 978-5-369-01195-9. - Текст: электронный. - URL: https://znanium.com/catalog/product/1860435 (дата обращения: 16.02.2023). – Режим доступа: по подписке.
4. Воронцова, Е. А. Программирование на С++ с погружением: практические задания и примеры кода - Москва: НИЦ ИНФРА-М, 2016. - 80 с. ISBN 978-5-16-105159-7. - Текст: электронный. - URL: https://znanium.com/catalog/product/563294 (дата обращения: 16.02.2023). – Режим доступа: по подписке.

**Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

1. Колдаев, В. Д. Структуры и алгоритмы обработки данных [Электронный ресурс]: учеб. пособие для вузов/ В. Д. Колдаев. - Москва: РИОР; Москва: ИНФРА-М, 2014. - 1 эл. опт. диск (CD-ROM), 294 с.: ил., табл. - (Высшее образование - бакалавриат). - Библиогр.: с. 285. - Лицензия до 23.06.2020 г. - ISBN 978-5-369-01264-2. - ISBN 978-5-16-009012-2: 15100.00 р. Имеются экземпляры в отделах /There are copies in departments: всего /all 2: ЭБС Кантиана(1), ч.з.N1(1) Свободны / free: ЭБС Кантиана(1), ч.з.N1(1)
2. Основы программирования. – режим доступа: http://www.intuit.ru/studies/courses/648/504/info
3. Видеолекции по курсу «Основы программирования». – режим доступа: http://www.youtube.com/watch?v=pxR3UoO9c9w
4. Сайт «Структуры и алгоритмы». – режим доступа: http://www.structur.h1.ru/
5. Структуры и алгоритмы обработки данных: Учебное пособие. – режим доступа: http://window.edu.ru/resource/820/44820

# Приложения

## Приложение 1

Код

## Приложение 2

Код